Indian J.Sci.Res. 17(2): 380 - 384, 2018 ISSN: 0976-2876 (Print)
ISSN: 2250-0138(Online)

FPGA IMPLEMENTATION AND DESIGN OF ARITHMETIC CODER

[1]A. Ravindar,[2]P. Bala krishna, [3]M. Asha Jyothi
[1,2,3]Department of Electronics and Communication Engineering, Keshav Memorial Institute of Technology, Narayanaguda, Hyderabad

Abstract-Set Partitioning in Hierarchical Trees (SPIHT) is a wavelet based compression algorithm that offers good compression ratios, a fully progressive bit-stream, and good image quality. This paper presents the results of adding arithmetic coding to the SPIHT images in the hopes of further reducing the image size. Arithmetic coding is a form of entropy encoding used in lossless data compression. Here 2D image coefficients after transformation are given as input to the arithmetic coder which will encode these coefficients in efficient way so that the memory requirement for storing will get reduced resulting in compressed size. For encoding the pixels are processed four in a group producing codeword after every 32 clock cycles. The design includes implementation of common bit detector, bit assembler and probability update parts.

***Keywords*:** Arithmetic coding, common bit detector, context model, set partitioning in hierarchical trees (SPIHT).

I. Introduction

Set Partitioning in Hierarchical Trees (SPIHT) is a wavelet-based image compression coder that offers a variety of good characteristics. These characteristics include:

• Good image quality with a high PSNR (peak to peak signal noise ratio)

• Fast coding and decoding

• Can be used for lossless compression

• Ability to code for exact bit rate or PSNR

The advantage of SPIHT is that it is fully progressive, meaning that we do not need the whole file to see the image. The image's PSNR(peak to peak signal to noise ratio) will be directly related to the amount of the file received from the transmitter. This means that our image quality will only increase with the percentage of the file received. After the SPIHT transformation some regularity will exist in the file. These regularities may allow us to further compress the file. With this in mind we investigated the addition of arithmetic compression to a SPIHT encoded image. A few different codeimplementations are presented here and the compression results are compared.

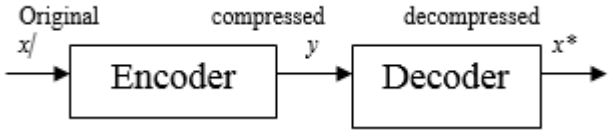

Fig1: Basic data compression

Lossless compression will be $x = x^*$ also called as entropy coding or reversible coding.

Lossy compression will be $x \neq x^*$ also called as irreversible coding.

A. Arithmetic Coder

Arithmetic coding (AC) is a special kind of entropy coding. Unlike Huffman coding, arithmetic coding doesn´t use a discrete number of bits for each symbol to compress. It reaches for every source almost the optimum compression in the sense of the Shannon theorem and is well suitable for adaptive models. The biggest drawback of the arithmetic coding is its low speed since of several needed multiplications and divisions for each symbol. The main idea behind arithmetic coding is to assign to each symbol an interval. Starting with the interval [0..1), each interval is divided in several subintervals, which sizes are proportional to the current probability of the corresponding symbols of the alphabet. The subinterval from the coded symbol is then taken as the interval for the next symbol. The output is the interval of the last symbol. Implementations write bits of this interval sequence as soon as they are certain. A fast variant of arithmetic coding, which uses less multiplications and divisions, is a range coder, which works byte oriented. The compression rate of a range coder is only a little bit less than pure arithmetic coding, and the difference in many real implementation is not noticeable.

This paper presents, for easier hardware implementation, a simple context model which is built on neighbor pixel's significant information is designed. Based on this context model, the context symbols which are formed by SPIHT algorithm are processed in parallel by the arithmetic coder in order to speed up the performance. Carry look-ahead circuits are used for cumulative probability update values in order to shorten the critical path.

The rest of the paper is organized as follows.

[1]Corresponding Author

Section II deals with related work and contributions. In section III, architecture of SPIHT, section IV deals with architecture of Arithmetic coder, section V deals with theimplementation results,section VI deals with theadvantages and applications of architecture, and section VII deals with conclusion.

II. Related Work And Contributions

W. A. Pearlman [2] proposed Set Partitioning in Hierarchical Trees (SPIHT) uses an AC method to improve its PSNR performance. The main feature of this coding method is that the ordering data is not explicitly transmitted instead it is based on the fact that the execution path of algorithm is defined by the results of comparison on its branching algorithm. So the encoding and decoding will have the same sorting algorithm, then the decoder can duplicate the encoders execution path if it receives the results of magnitude comparison and the ordering information can be recovered from execution path. The algorithm for sorting all coefficients are $2^n<=|c_{i,j}|<2^{n+1}$,with n decrements in each pass. Given if $|c_{i,j}|>=2^n$ then the co-efficient is significant otherwise insignificant.

D Taubman [3] proposed a new architecture for image compression algorithm. It describes a novel image compression algorithm known as EBCOT: Embedded Block Coding with Optimized Truncation. The EBCOT algorithm is related in various degrees to easier work on scalable image compression. The EBCOT algorithm uses a wavelet transform to generate the subband samples which are to be optimized and coded. EBCOT algorithm generates scalable compressed bit streams, whereas its predecessors typically exhibited one (SNR) or at most two(SNR and resolution) dimensions of scalability, the EBCOT algorithm produces bit streams which are SNR and resolution scalable and which also have a random access attribute where by independent portions of the bit streams corresponds to different spatial regions of the image. A key scalable compression is that the target bit rate or reconstruction resolution need not be known at the of compression. A related advantage is that the image need not be compressed multiple times in order to achieve a target bit-rate. Rather than focusing on generating a single scalable bit-stream to represent the entire image, EBCOT partitions each subband into relatively small blocks of samples and generates a separate highly scalable bit-stream to represent each so-called code-block, B_i. The bit-stream associated with B_i may be independently truncated to any of a collection of different lengths, R_i^n where the increase in reconstructed image distortion resulting from these truncations is given by D_i^n, with respect to an appropriate distortion metric.The EBCOT algorithm offers state-of-the-art compression performance with a rich set of bit stream features, including resolution scalability, SNR scalability and random access property.

EBCOT makes two significances, the generation of finely embedded bit streams and it introduces abstract quality layers which are not directly related to the structural properties of entropy coders.

Wiseman [4] proposed hardware architecture for quasi AC which is simple version of AC. It is a combination of parallel software and pipeline hardware which can give very fast compression withoutdecline of compression efficiency. Quasi Arithmetic Coding is a version of well-known arithmetic compression algorithm, but the compression is done in software. The part of hardware is developed as pipeline hardware. The software part can also be implemented in parallel mode. Quasi arithmetic coding is one of the methods which approximate AC by using state tables. The architecture uses pipeline processing to compute each stage of AC, which eliminate internal high frequency clock and utilizes fast look up table for state transitions and it cannot support for multi-context AC processing in image compression fields.

K. Andra, T. Acharya and C. Chakrabarti [5] proposed a new methodology for binary arithmetic coding which reduces the number of arithmetic operations significantly at the expense of mild reduction in the compression ratio which makes this technique particularly suitable for low power implementation in both software and hardware. This method consists of two symbol non overlapping window and moving majority of computations to Least Probable Symbol path. As a result, it reduces the addition/subtraction required by 60-70% with loss in compression ratio about 1-3% compared with Q-coder where the dynamic allocation of More Probable Symbol or Least Probable Symbol is done based on context.

W. B. Huang, A. W. Y. Su and Y. H. Kuo [6] proposed architecture of SPIHT that has a straight forward procedure and requires no tables. This makes SPIHT a more appropriate algorithm for lower cost hardware implementation. The modifications for SPIHT algorithm include a simplification of co-efficient scanning process. A 1-D addressing method instead of original 2-D arrangement of wavelet coefficients and a fixed memory allocation for data lists instead of dynamic allocation approach required in original SPIHT. It provides an extremely fast throughput and easier hardware implementation but the distortion is slightly increased. Here no look up table is essential.

D. Marpe, H. Schwarz and T. Wiegand [7] proposed a new adaptive entropy coding scheme for video compression is presented, a novel approach for coding of transform coefficients and a table look up method for probability estimation and arithmetic coding. It is a three step process, first a one bit symbol called CBP4 is transmitted for each block of transform coefficients unless the coded block pattern(CBP) on macro block level indicates that regarded block has non zero coefficients.

The CBP4 symbol is set to one, if there are any significant, i.e., non zero coefficients inside the corresponding block. If it is zero no further information is transmitted for the block, otherwise, in a second coding step, a significant map specifying the position of significant co-efficient is encoded. Finally, the absolute value as well as the sign is encoded for each significant transform co-efficient. These values are transmitted in reverse scanning order. For coding of interlaced material, average bit rate savings are in the range from 15-27% in comparison to UVLC. St the expense of minor increase in bit-rate, a table-based low-complexity arithmetic coding engine is incorporated in H.26L.

. W. Wheeler and W. A. Pearlman [8] has proposed a variant of SPIHT image compression algorithm called no list SPIHT (NLS). NLS operates without linked list and suitable for fast and simple hardware implementation. Instead of list, a state table with four bits per coefficient keeps track of set partition and what information has been encoded. NLS uses same set structures and partitioning rules as SPIHT. The trees are tested for significance breadth first. Significance tests are made in a different order than SPIHT because SPIHT performs significance tests roughly breadth first while NLS performs tests strictly breadth first. NLS and SPIHT produce bit streams with the same bits in the different order. SPIHT's ordering offers slightly better performance at certain points in the bit stream during set significance passes. No arithmetic coding was introduced at the significance test or any symbol produced by NLS algorithms. Back-end arithmetic coding using context and joint encoding improves the performance.

J. Bac and V. K. Prasanna [9] proposed architecture for embedded zero tree wavelet(EZW) algorithm, partitioning and mapping of the computation leads to parallel operation of the independent processor and balances the workload. Data mapping technique specifies the memory access and the processor architecture. Video codec based on the parallel architecture can simultaneously handle multiple video channel of various resolutions. The EZW algorithm is based on hypothesis that if a wavelet coefficient at a coarse scale is significant with respect to a given threshold T, then all wavelet coefficients of the same orientations in the same spatial location at finer scales are likely to be insignificant with respect to T. This algorithm generates fully embedded code from the wavelet coefficients 2D-DWT. The architecture can be used for a low power design of mobile/visual communication system by increasing the parallelism and thereby reducing the system clock frequency.

III. Spiht Architecture

The architecture of SPIHT image compression is shown in Fig 2

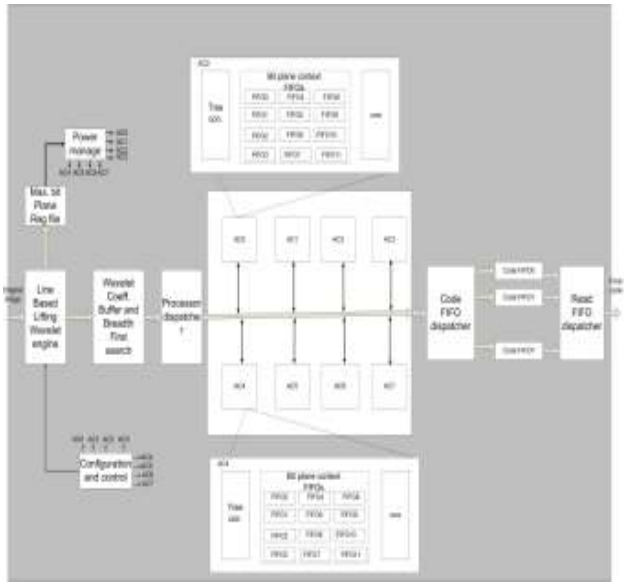

Fig 2: Architecture of SPIHT

The arithmetic coder consists of three main parts, the tree construction part noted as tree con, the tree plane context FIFOs array and coding core. The tree construction part will visits the coefficients in breadth first order, for speed up purpose all valid bit planes are processed in parallel and the invalid bit planes are idle by stopping the corresponding clock. The bit plane context FIFOs arrays have twelve FIFOs store the context values of bit plane. The coding core part will calculate the context value from code byte.

IV. Architecture Of Arithmetic Coder

Each coder core is having four cores whose architecture is identical and work independently which allows multiple contexts to be processed parallel. Fig 3 shows the architecture of single core of Arithmetic coder.

Here in each core the read context and symbol part will compare the context label with its internal register to know whether the context label confirms to its own tag or not, if yes then it will be sent to boundary update part otherwise discarded. The lower bound update and upper bound update parts will calculate the new bound values, the new update values are transmitted to the common bit detector part. For symbol probability the values are assigned randomly and can be changed. The symbols arriving at the read context and symbol part we are giving symbol probability for updating intervals. The cumulative probability part will be having zero initially, the values are updated based on the value of write address from control logic part. The common bit detector part will unroll the internal loop and records the bits from MSB to LSB between two registers new_high and new_low. At the end the same bits are collected to form byte align code through bit assembly part then finally these bytes are transmitted to

code stream output part for emitting these parts to external bus. The control unit is responsible for running of code and sending various commands and control signals. The probability update parts will use old_low and old_high values to update the probability interval. After every 32 clock cycles, the code word is generated.

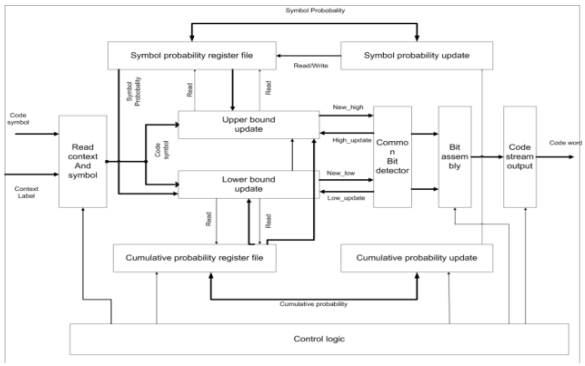

Fig 3: Architecture of Arithmetic Coder

V. System Results

The arithmetic coder for encoding and four parallel cores for compression are programmed using verilog language. Verilog has built-in gate level and transistor level primitives, it is much better than VHDL at below RTL level.The coder and the cores are simulated using the simulation tool modelsim 6.2. The results in terms of numbers and waveforms are analysed. One sample window showing simulation results forcoder in Fig 5 and cores in Fig 6.

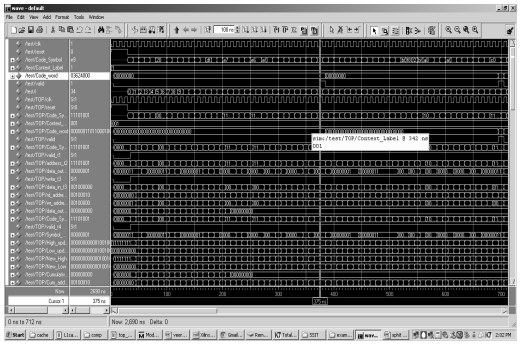

Fig 5: simulation waveform for Arithmetic Coder.

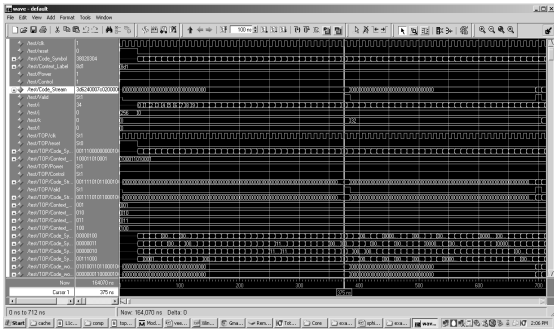

Fig 6: simulation waveform for Cores.

Then the code is synthesized using Xilinx software which outputs the maximum frequency about 43.843 MHz, minimum period 22.808 ns. Then the synthesized code is tested on Spartan 3AN FPGA kit.

VI. Advantages And Applications

Encoding plays a very important role in processing issues. By encoding image or text one can secure the data and which requires less memory for storage which in turn leads for faster transfer of data.

The hardware part of arithmetic coding can be incorporated in computer applications where encoding is crucial for authenticating an individual who is trying to access the services.

VII. Conclusion

This system design has been implemented the Arithmetic Coder architecture used in SPIHT algorithm to encode the image pixels in efficient way which in turn leads to compression of the image. By the implementation of the above said algorithm, the speed will be increased by pipeline processing at the arithmetic coder and by duplicating the architecture of core four times.

References

[1]. Kai Liu, Evgeniy Belyaev, and Jie Guo, "VLSI Architecture of Arithmetic Coder Used in SPIHT"IEEE transactions on very large scale integration (VLSI) systems, vol. 20, no. 4, pp. 697-710, april2012.

[2]. Said and W. A. Pearlman, "A new ,fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 6, no. 3, pp. 243–249, Mar. 1996.

[3]. D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.

[4]. Y. Wiseman, "A pipeline chip for quasi arithmetic coding," *IEICE Trans. Fundamentals*, vol. E84-A, no. 4, pp. 1034–1041, Apr. 2001.

[5]. K. Andra, T. Acharya, and C. Chakrabarti, "A multi-bit binary arithmetic coding technique," in *Proc. Int. Conf. Image Process.*, Vancouver, BC, Canada, Sep. 2000, vol. 1, pp. 928–931.

[6]. W.-B. Huang, A. W. Y. Su, and Y.-H. Kuo, "VLSI implementation of a modified efficient SPIHT encoder," *IEICE Trans. Fundamentals*, vol. E89-A, no. 12, pp. 3613–3622, Dec. 2006.

[7]. D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE*

Trans. Circuits Syst. for Video Technol., vol. 13, no. 7, pp.620–636, Jul. 2003.

[8]. F. W. Wheeler and W. A. Pearlman, "SPIHT image compression without lists," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Istanbul, Turkey, Jun. 2000, pp. 2047–2050.

[9]. J. Bac and V. K. Prasanna, "A fast and area-efficient VLSI architecture for embedded image coding," in *Proc. Int. Conf. Image Process.*, Oct. 1995, vol. 3, pp. 452–455.